# BTCS for the 1D Heat Equation, in a Nutshell

gerry@pdx.edu

ME 448/548
Winter 2020

1. Combine finite difference approximations for $\partial u/\partial t$ at $x = x_i$

$$\left.\frac{\partial u}{\partial t}\right|_{t_k, x_i} = \frac{u_i^k - u_i^{k-1}}{\Delta t} + \mathcal{O}(\Delta t) \tag{1}$$

and $\partial^2 u/\partial x^2$ at time $t_k$

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{t_k, x_i} = \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + \mathcal{O}(\Delta x^2). \tag{2}$$

to get

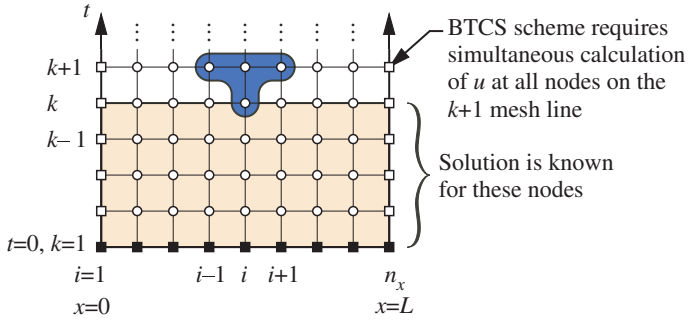
$$\frac{u_i^k - u_i^{k-1}}{\Delta t} = \alpha \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2) \tag{3}$$

Drop the truncation error terms, shift the $k$ subscripts by one, and move all $u^{k+1}$ terms to the left hand side

$$\left[-\frac{\alpha}{\Delta x^2}\right] u_{i-1}^{k+1} + \left[\frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2}\right] u_i^{k+1} + \left[-\frac{\alpha}{\Delta x^2}\right] u_{i+1}^{k+1} = \frac{1}{\Delta t} u_i^k \tag{4}$$

Equation (4) is the computational formula for the BTCS scheme. It is an *implicit* scheme because all $u^{k+1}$ values are coupled and must be updated simultaneously.

2. Computational Molecule




3. The BTCS method is *unconditionally stable* for the heat equation.

   The benefit of stability comes at a cost of increased complexity of solving a linear system of equations at each time step.

4. For the one-dimensional heat equation, the linear system of equations for the BTCS method can be organized into a tridiagonal matrix.

$$\begin{bmatrix} a_1 & b_1 & 0 & 0 & 0 & 0 \\ c_2 & a_2 & b_2 & 0 & 0 & 0 \\ 0 & c_3 & a_3 & b_3 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & c_{n_x-1} & a_{n_x-1} & b_{n_x-1} \\ 0 & 0 & 0 & 0 & c_{n_x} & a_{n_x} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n_x-1} \\ u_{n_x} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n_x-1} \\ d_{n_x} \end{bmatrix} \tag{5}$$

where the coefficients of the matrix are stored compactly as vectors

$$a_i = (1/\Delta t) + (2\alpha/\Delta x^2), \qquad b_i = c_i = -\alpha/\Delta x^2, \qquad d_i = (1/\Delta t)u_i^k. \tag{6}$$

This system of equations is efficiently solved with a form of LU factorization. The LU factors need to be computed only once before the first time step.

5. MATLAB implementation: code from `demoBTCS`

```
% --- Coefficients of the tridiagonal system
b = (-alfa/dx^2)*ones(nx,1);     %  Super diagonal: coefficients of u(i+1)
c = b;                           %  Subdiagonal:    coefficients of u(i-1)
a = (1/dt)*ones(nx,1) - (b+c);   %  Main Diagonal:  coefficients of u(i)
a(1) = 1;     b(1) = 0;          %  Fix coefficients of boundary nodes
a(end) = 1;   c(end) = 0;
[e,f] = tridiagLU(a,b,c);        %  Save LU factorization

% --- Assign IC and save BC values in ub.  IC creates u vector
x = linspace(0,L,nx)';   u = sin(pi*x/L);   ub = [0 0];

% --- Loop over time steps
for k=2:nt
  d = [ub(1); u(2:nx-1)/dt;  ub(2)];   %  Update RHS, preserve BC
  u = tridiagLUsolve(e,f,b,d);         %  Solve the system
end
```

A more general implementation is in `heatBTCS`.

6. The `demoBTCS` function demonstrates the correct behavior of truncation error as $\Delta x$ and $\Delta t$ are reduced.

```
>> convBTCS
```

| nx | nt | error | E(j)/E(j-1) | p |
|---|---|---|---|---|
| 4 | 5 | 5.346e-02 | 0.0000 | 0.0000 |
| 8 | 21 | 1.186e-02 | 0.2219 | 2.1723 |
| 16 | 92 | 2.716e-03 | 0.2290 | 2.1268 |
| 32 | 386 | 6.522e-04 | 0.2401 | 2.0581 |
| 64 | 1589 | 1.594e-04 | 0.2444 | 2.0326 |
| 128 | 6453 | 3.939e-05 | 0.2471 | 2.0168 |
| 256 | 26012 | 9.790e-06 | 0.2485 | 2.0084 |
| 512 | 104452 | 2.440e-06 | 0.2493 | 2.0042 |

| nx | nt | error | E(j)/E(j-1) | p |
|---|---|---|---|---|
| 4 | 4 | 6.444e-02 | 0.0000 | 0.0000 |
| 8 | 8 | 2.737e-02 | 0.4248 | 1.2353 |
| 16 | 16 | 1.276e-02 | 0.4661 | 1.1014 |
| 32 | 32 | 6.172e-03 | 0.4838 | 1.0475 |
| 64 | 64 | 3.037e-03 | 0.4921 | 1.0230 |
| 128 | 128 | 1.507e-03 | 0.4961 | 1.0113 |
| 256 | 256 | 7.504e-04 | 0.4981 | 1.0056 |
| 512 | 512 | 3.745e-04 | 0.4990 | 1.0028 |